UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/751,761	12/29/2000	Ronald D. Smith	2207/10119	5065

7590 06/26/2006

Kenyon & Kenyon
Suite 600
333 W. San Carlos Street
San Jose, CA 95110-2711

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 06/26/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/751,761	SMITH, RONALD D.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 April 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *20-38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *20-38* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *10 May 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 20-38 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Extension of Time and Amendment as received on 4/24/2006.

Claim Objections

3. Claim 28 recites the limitation "the execution unit" in line 2. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 20, 24, 26, 33, and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Swoboda et al., U.S. Patent No. 6,643,803 (as applied in the previous Office Action and herein referred to as Swoboda) in view of Ehlig et al., U.S Patent No. 5,551,050 (herein referred to as Ehlig).

6. Referring to claim 20, Swoboda has taught a method comprising:

a) detecting a stall in an execution stage of a processor core. See claim 1 and the abstract, and note that a bubble/stall is detected.

b) generating a neutral instruction. See claim 1 and the abstract, and note that a system resource access is generated. And, the instruction is neutral because a resource can be read as a result of the instruction. Reading a resource does not modify the architectural state of the processor, and therefore, a read is a neutral instruction.

c) providing said neutral instruction to said execution stage. See claim 1 and the abstract, and note that the system resource access is jammed into the bubble, which is in the pipeline.

d) executing said neutral instruction to ascertain an architectural state value for said processor core. See claim 1 and the abstract, and note that the access is executed and, as a result, a value is obtained.

e) Swoboda has not taught comparing said architectural state value for said processor core to an architectural state value for a second processor core. However, Ehlig has taught the concept of comparing first processor data obtained from emulation circuitry to second processor data obtained from emulation circuitry. See column 6, lines 7-22. As Ehlig explains, this type of comparison is useful in redundant processing applications which vote. As is known, redundancy and voting allows for increased fault tolerance within the system. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Swoboda such that at least one redundant processor is implemented and the results of the processors are compared. As explained, one would be motivated to make such a combination to increase fault tolerance.

7. Referring to claim 24, Swoboda has taught a method as described in claim 20. Swoboda has further taught that the execution of said neutral instruction causes said processor core to access a value stored in a register in said processor core. From the abstract it is disclosed that system resources, which include registers (column 2, lines 50-51), are read when a neutral instruction is executed.

8. Referring to claim 26, Swoboda has taught a system comprising:

a) stall logic coupled to an execution stage of a processor core to detect a stall in said execution. See claim 1 and the abstract, and note that a bubble/stall is detected.

b) comparison logic coupled to said execution stage, wherein upon occurrence of the stall said execution stage is to execute a neutral instruction to ascertain an architectural state value for said processor core. See claim 1 and the abstract, and note that, in response to a bubble/stall, a system resource access is jammed into the bubble and executed, thereby avoiding additional pipeline delay. A system resource includes registers (column 2, lines 50-51). Since registers hold values, these jammed instructions would ascertain (read) values for the processor. In addition, it should be realized that these instructions are used for testing purposes, as described in column 2, lines 46-65. Therefore, for a test to occur, comparison logic must inherently exist in order to determine whether the test was a failure or success. There must be an expected outcome of some sorts which would be compared with the outcome obtained from executing the neutral instruction.

c) Swoboda has not taught comparing said architectural state value for said processor core to an architectural state value for a second processor core. However, Ehlig has taught the concept of comparing first processor data obtained from emulation circuitry to second processor data

obtained from emulation circuitry. See column 6, lines 7-22. As Ehlig explains, this type of comparison is useful in redundant processing applications which vote. As is known, redundancy and voting allows for increased fault tolerance within the system. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Swoboda such that at least one redundant processor is implemented and the results of the processors are compared. As explained, one would be motivated to make such a combination to increase fault tolerance.

9. Referring to claim 33, Swoboda has taught a set of instructions residing in a storage medium (Fig.12, and note the instruction memory on the far left - "INST MEM"), said set of instructions capable of being executed in an execution stage by a processor core for implementing a method to test the processor core, the method comprising:

a) detecting a stall in an execution stage of a processor. See claim 1 and the abstract, and note that a bubble/stall is detected.

b) generating a neutral instruction. See claim 1 and the abstract, and note that a system resource access is generated. And, the instruction is neutral because a resource can be read as a result of the instruction. Reading a resource does not modify the architectural state of the processor, and therefore, a read is a neutral instruction.

c) providing said neutral instruction to said execution stage. See claim 1 and the abstract, and note that the system resource access is jammed into the bubble, which is in the pipeline.

d) executing said neutral instruction to ascertain an architectural state value for said processor core. See claim 1 and the abstract, and note that the access is executed and a s a result, a value is obtained.

e) Swoboda has not taught comparing said architectural state value for said processor core to an architectural state value for a second processor core. However, Ehlig has taught the concept of comparing first processor data obtained from emulation circuitry to second processor data obtained from emulation circuitry. See column 6, lines 7-22. As Ehlig explains, this type of comparison is useful in redundant processing applications which vote. As is known, redundancy and voting allows for increased fault tolerance within the system. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Swoboda such that at least one redundant processor is implemented and the results of the processors are compared. As explained, one would be motivated to make such a combination to increase fault tolerance.

10. Referring to claim 37, Swoboda has taught a set of instructions as described in claim 33. Swoboda has further taught that in said method the execution of said neutral instruction causes said processor core to access a value stored in a register in said processor core. From the abstract it is disclosed that system resources, which include registers (column 2, lines 50-51), are read when a neutral instruction is executed.

11. Claims 21, 27, and 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Swoboda in view of Ehlig and further in view of Sato, U.S. Patent No. 5,903,768 (as applied in the previous Office Action).

12. Referring to claim 21, Swoboda in view of Ehlig has taught a method as described in claim 20. Swoboda in view of Ehlig has not taught that said neutral instruction is generated when a plurality of instructions are generated by a compiler. However, Sato has taught such a

concept. More specifically, in column 2, lines 4-13, Sato discloses that a compiler is used to generate instructions and the order in which they are executed. Furthermore, when a hazard between two instructions cannot be eliminated, the compiler inserts a NOP instruction between them to overcome the hazard. This is equivalent to generating a neutral instruction because the neutral instruction is the same as a NOP in the sense that it does not affect the architectural state of the processor. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Swoboda in view of Ehlig such that a neutral instruction is generated when a plurality of instructions are generated by the compiler. This would be obvious because Sato has taught the known concept of overcoming a hazard by generating NOPs (or neutral instructions), and hazards must be overcome in order to prevent data corruption.

13. Referring to claim 27, Swoboda in view of Ehlig has taught a system as described in claim 26. Furthermore, claim 27 is rejected for the same reasons set forth in claim 21.

14. Referring to claim 34, Swoboda in view of Ehlig has taught a set of instructions as described in claim 33. Furthermore, claim 34 is rejected for the same reasons set forth in claim 21.

15. Claims 22-23, 25, 28-30, 35-36, and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Swoboda in view of Ehlig and further in view of Mandyam et al., U.S. Patent No. 6,285,974 (as applied in the previous Office Action and herein referred to as Mandyam).

16. Referring to claim 22, Swoboda in view of Ehlig has taught a method as described in claim 20. Swoboda in view of Ehlig has not taught that said neutral instruction is generated by a No-operation (NOP) pseudo-random generator. However, Mandyam has taught generating test

instructions using a random test generator. A person of ordinary skill in the art would have recognized that by implementing a random generator to generate instructions, sources of bias are eliminated. Consequently, truly random instructions may be generated which would allow for the possibility of testing any register at any appropriate point within the execution. As a result, in order to perform random testing, as opposed to biased testing, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Swoboda in view of Ehlig such that a No-operation (neutral instruction) pseudo-random generator is used to generate neutral instructions, as taught by Mandyam.

17. Referring to claim 23, Swoboda in view of Ehlig and further in view of Mandyam has taught a method as described in claim 22. Swoboda has further taught that the execution of said neutral instruction causes said processor core to access a value stored in a register in said processor core. From the abstract it is disclosed that system resources, which include registers (column 2, lines 50-51), are read when a neutral instruction is executed.

18. Referring to claim 25, Swoboda in view of Ehlig has taught a method as described in claim 20. Swoboda in view of Ehlig has not taught that said neutral instruction is generated by a post-processor device. However, Mandyam has taught such a concept. See Fig.3 and column 6, lines 10-14. Note that a post-processor is used to generate instructions which are used to perform a self-check. This allows for detection of architectural violations as described in column 6, lines 10-25. Since Mandyam has taught that test instructions may be generated by a post-processor, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement a post-processor in Swoboda in view of Ehlig for such a purpose.

19. Referring to claim 28, Swoboda in view of Ehlig has taught a system as described in claim 26. Furthermore, claim 28 is rejected for the same reasons set forth in claim 22.

20. Referring to claim 29, Swoboda in view of Ehlig has taught a system as described in claim 28. Furthermore, claim 29 is rejected for the same reasons set forth in claim 23.

21. Referring to claim 30, Swoboda in view of Ehlig and further in view of Mandyam has taught a system as described in claim 29. Swoboda in view of Ehlig and further in view of Mandyam has not taught that said neutral instruction includes ORing the contents of said register with itself. However, an OR operation is well known and expected in the art. And, it is known that ORing an operand with itself is a neutral operation as ORing 0 and 0 yields 0 and ORing 1 and 1 yields 1. Since an OR operation is a fundamental logic operation, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement the function of ORing the contents of a register with itself.

22. Referring to claim 35, Swoboda in view of Ehlig has taught a set of instructions as described in claim 33. Furthermore, claim 35 is rejected for the same reasons set forth in claim 22.

23. Referring to claim 36, Swoboda in view of Ehlig has taught a set of instructions as described in claim 35. Furthermore, claim 36 is rejected for the same reasons set forth in claim 23.

24. Referring to claim 38, Swoboda in view of Ehlig has taught a set of instructions as described in claim 33. Furthermore, claim 38 is rejected for the same reasons set forth in claim 25.

25. Claims 31-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Swoboda in view of Ehlig in view of Mandyam and further in view of Hennessy and Patterson, Computer Organization and Design, 2nd Edition, 1998 (as applied in the previous Office Action and herein referred to as Hennessy).

26. Referring to claim 31, Swoboda in view of Ehlig and further in view of Mandyam has taught a system as described in claim 29. Swoboda in view of Ehlig and further in view of Mandyam has not taught that said neutral instruction includes ANDing the contents of said register with all binary 1 values. However, an AND operation is well known and expected in the art, and supported by Hennessy. See pages 225-226. Hennessy has taught that each resulting bit will be 1 only if both corresponding operand bits are 1. And, an operand of an AND operation can be an operand of all 0's, an operand of all 1's, and everything in between. Masking (ANDing operation) is used to isolate fields, which in turn allows for the examination of bits within a word. Consequently, since an AND operation is a fundamental logic operation, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement the function of ANDing the contents of a register with all binary 1 values, as taught by Hennessy.

27. Referring to claim 32, Swoboda in view of Ehlig and further in view of Mandyam has taught a system as described in claim 29. Swoboda in view of Ehlig and further in view of Mandyam has not taught that said neutral instruction includes ORing the contents of said register with all binary 0 values. However, an OR operation is well known and expected in the art, and supported by Hennessy. See pages 225 and 227. Hennessy has taught that each resulting bit will be 1 if either one of the corresponding operand bits are 1. And, an operand of an OR operation can be an operand of all 0's, an operand of all 1's, and everything in between. Consequently,

since an OR operation is a fundamental logic operation, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement the function of ORing the contents of a register with all binary 0 values, as taught by Hennessy.

Conclusion

28. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

DJH
David J. Huisman
June 20, 2006

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100